Amendments to the Claims:

Please cancel claims 22, 23, 24, 26, 29, 30, 31, 32, 33, 57-61, 64-68, and 80. Please also amend claims 1, 2, 36, 37, 71 and 72 as indicated in the listing of claims, below. This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) An apparatus comprising:

a translation lookaside buffer (TLB) in a processor having a plurality of TLB entries, each TLB entry being associated with a virtual machine extension (VMX) tag word indicating if the associated TLB entry is invalidated according to the processor mode when an invalidation operation is performed, the processor mode being one of execution in a virtual machine (VM) and execution not in a virtual machine, the invalidation operation belonging to a non-empty set of invalidation operations composed of a union of (1) a possibly empty set of operations that invalidate a variable number of TLB entries, (2) a possibly empty set of operations that invalidate exactly one TLB entry, (3) a possibly empty set of operations that invalidate the plurality of TLB entries, (4) a possibly empty set of operations that enable and disable use of virtual memory, and (5) a possibly empty set of operations that configure physical address size, page size or other virtual memory system behavior in a manner that changes the manner in which a physical machine interprets the TLB entries[.];

wherein the invalidation operations include IA-32-specific operations (a) IA-32 task switches involving changes of virtual memory related control

registers, or (b) loading of control registers that modify IA-32-specific page size extension (PSE) and physical address extension (PAE).

2. (Currently amended) The apparatus of claim 1 wherein the invalidation operation is one of (1) a loading of a first control register conditioned on a global bit, (2) an execution of a page invalidate instruction, (3) an IA-32 task switch involving change of at least one virtual memory related control register ~~a task switch that modifies the first control register,~~ (4) a loading of a second control register that modifies one of a protected mode indicator and a page mode indicator, and (5) a loading of a third control register that modifies one of an IA-32-specific page size extension (PSE), a page global enable (PGE), and a physical address extension (PAE).

3. (Original) The apparatus of claim 2 wherein the processor is in or not in VMX mode and the TLB entry is not invalidated at loading of the first control register when one of a transition into VMX mode (a VM entrance) and a transition out of VMX mode (a VM exit) occurs.

4. (Original) The apparatus of claim 3 wherein the VMX tag word is a single bit and

the VMX tag word is negated for a new TLB entry when the processor is not in VMX mode and the VMX tag word is asserted for a new TLB entry when the processor is in VMX mode; and

the TLB entry is invalidated when an invalidation operation is performed and the VMX tag word is asserted and the processor is in VMX mode.

5. (Original) The apparatus of claim 4 wherein the TLB entry is invalidated irrespective of value of the VMX tag word when an invalidation operation is performed and the processor is not in VMX mode.

6. (Original) The apparatus of claim 4 wherein a field in a control register is designated the translation lookaside buffer virtual machine extension (TLBVMX) word and the TLB entry is invalidated when an invalidation operation is performed and the VMX tag word matches the TLBVMX word and the processor is not in VMX mode.

7. (Original) The apparatus of claim 3 wherein the VMX tag word is a single bit and

the VMX tag word is asserted for a new TLB entry when the processor is not in VMX mode and the VMX tag word is negated for a new TLB entry when the processor is in VMX mode; and

the TLB entry is invalidated when an invalidation operation is performed and the VMX tag word is negated and the processor is in VMX mode.

8. (Original) The apparatus of claim 7 wherein the TLB entry is invalidated irrespective of the value of the VMX tag word when an invalidation operation is performed and the processor is not in VMX mode.

9. (Original) The apparatus of claim 7 wherein a field in a control register is designated the TLBVMX word and the TLB entry is invalidated when an invalidation operation is performed and the VMX tag word matches the TLBVMX word and the processor is not in VMX mode.

10. (Original) The apparatus of claim 1 wherein invalidation of a TLB entry by an invalidation operation is further conditioned upon value of one or more control words associated with the TLB.

11. (Original) The apparatus of claim 10 wherein the control word or words associated with the TLB are located in one or more of the control registers of the processor or in a Virtual Machine Control Structure (VMCS) in memory.

12. (Original) The apparatus of claim 11 wherein one of the control words associated with the TLB is distinguished such that the VMX tag word is set to match a distinguished control word for a new TLB entry, the distinguished control word associated with the TLB being designated the TLBVMX word.

13. (Original) The apparatus of claim 12 wherein the TLBVMX word is set to one of a plurality of values that constitute a proper subset of a set of all possible values for a VMX tag word when the processor mode corresponds to execution in a virtual machine (VM).

14. (Original) The apparatus of claim 13 wherein the TLB entry is invalidated when an invalidation operation is performed and the value of the associated VMX tag word matches the value of the TLBVMX word and the processor mode corresponds to execution in a virtual machine (VM).

15. (Original) The apparatus of claim 14 wherein the one or more control words associated with the TLB, including the TLBVMX word, are configurable when the processor mode corresponds to execution not in a virtual machine.

16. (Original) The apparatus of claim 13 wherein the TLBVMX word is set to one of a plurality of values that constitute a proper subset of a set of all possible values for a VMX tag word when the processor mode corresponds to execution not in a virtual machine.

17. (Original) The apparatus of claim 16 wherein a set theoretic intersection of the plurality of values allowable for the TLBVMX word when the processor mode corresponds to execution not in a virtual machine and the plurality of values allowable for the TLBVMX word when the processor mode corresponds to execution in a virtual machine (VM) is an empty set.

18. (Original) The apparatus of claim 16 wherein invalidation of a TLB entry by an invalidation operation is further conditioned upon value of a second control word associated with the TLB when the processor mode corresponds to execution not in a virtual machine.

19. (Original) The apparatus of claim 18 wherein the TLB entry is invalidated when an invalidation operation is performed and the associated VMX tag word matches the second control word associated with the TLB and the processor mode corresponds to execution not in a virtual machine.

20. (Original) The apparatus of claim 18 wherein the TLB entry is invalidated when an invalidation operation is performed and the logical AND of the associated VMX tag word and the second control word associated with the TLB matches the TLBVMX word and the processor mode corresponds to execution in a virtual machine, the second control word associated with the TLB being designated the TLBVMX mask word.

21. (Original) The apparatus of claim 20 wherein access by software to configure a portion of the TLBVMX word when the processor mode corresponds to execution in a virtual machine (VM) is conditioned upon value of the TLBVMX mask word such that software executing when the processor mode corresponds to execution in a virtual machine (VM) is able to set the TLBVMX word to a value such that the logical AND of a new value of the TLBVMX word and a value of the TLBVMX mask word matches a logical AND of a previous value of the TLBVMX word and the value of TLBVMX mask word.

22. Cancel.

23. Cancel.

24. Cancel.

25. (Amended) The apparatus of claim [24] 20 wherein the bits configured in the VMX tag word and the TLBVMX word are determined by an execution of a specified processor instruction in a specified manner.

26. Cancel.

27. (Original) The apparatus of claim 18 wherein the TLB entry is invalidated when an invalidation operation is performed and the logical AND of the associated VMX tag word with a logical NOT of the second control word associated with the TLB matches the TLBVMX word and the processor mode corresponds to execution in a virtual machine (VM), the second control word associated with the TLB being designated the TLBVMX inverted mask word.

28. (Original) The apparatus of claim 27 wherein access by software to configure a portion of the TLBVMX word when the processor mode corresponds to execution in a virtual machine (VM) is conditioned upon the value of the TLBVMX inverted mask word such that software executing when the processor mode corresponds to execution in a virtual machine (VM) is able to set the TLBVMX word to a value such that a logical AND of a new value of the TLBVMX word and the logical NOT of the TLBVMX inverted mask word matches the logical AND of a previous value of the TLBVMX word and the logical NOT of TLBVMX inverted mask word.

29. Cancel.

30. Cancel.

31. Cancel.

32. Cancel.

33. Cancel.

34. (Original) The apparatus of claim 16 wherein size of a VMX tag word and the TLBVMX word is determined by executing a specified processor instruction in a specified manner.

35. (Original) The apparatus of claim 34 wherein the processor is compatible with the Intel Architecture and the specified instruction is a CPUID instruction and the specified manner is to have a specified value in an EAX register when the CPUID instruction is executed.

36. (Currently Amended) A method comprising:

associating a translation lookaside buffer (TLB) entry in a plurality of TLB entries in a processor with a virtual machine extension (VMX) tag word to indicate if the associated TLB entry is invalidated according to the processor mode when an invalidation operation is performed, the processor mode being one of execution in a virtual machine (VM) and execution not in a virtual machine; and

performing the invalidation operation, the invalidation operation belonging to a non-empty set of invalidation operations composed of a union of (1) a possibly empty set of operations that invalidate a variable number of TLB entries, (2) a possibly empty set of operations that invalidate exactly one TLB entry, (3) a possibly empty set of operations that invalidate the plurality of TLB entries, (4) a possibly empty set of operations that enable and disable use of virtual memory, and (5) a possibly empty set of operations that configure physical address size, page size or other virtual memory system behavior in a manner that changes the manner in which a physical machine interprets the TLB entries;

wherein the invalidation operations include IA-32-specific operations (a) IA-32 task switches involving changes of virtual memory related control registers, or (b) loading of control registers that modify IA-32-specific page size extension (PSE) and physical address extension (PAE).

37. (Currently Amended) The method of claim 36 wherein performing comprises performing the invalidation operation being one of (1) a loading of a first control register conditioned on a global bit, (2) an execution of a page invalidate instruction, (3) ~~a task switch that modifies the first control register~~ an IA-32 task switch involving change of at least one virtual memory related control register, (4) a loading of a second control register that modifies one of a protected

mode indicator and a page mode indicator, and (5) a loading of a third control register that modifies one of an IA-32-specific page size extension (PSE), a page global enable (PGE), and a physical address extension (PAE).

38. (Original) The method of claim 37 wherein the processor is in or not in VMX mode and the TLB entry is not invalidated at loading of the first control register when one of a transition into VMX mode (a VM entrance) and a transition out of VMX mode (a VM exit) occurs.

39. (Original) The method of claim 38 further comprising:

negating the VMX tag word for a new TLB entry when the processor is not in VMX mode, the VMX tag word being a single bit;

asserting the VMX tag word for a new TLB entry when the processor is in VMX mode; and

invalidating the TLB entry when an invalidation operation is performed and the VMX tag word is asserted and the processor is in VMX mode.

40. (Original) The method of claim 39 wherein invalidating comprises invalidating the TLB entry irrespective of value of the VMX tag word when an invalidation operation is performed and the processor is not in VMX mode.

41. (Original) The method of claim 39 wherein invalidating comprises invalidating the TLB entry when an invalidation operation is performed and the VMX tag word matches the translation lookaside buffer virtual machine extension (TLBVMX) word and the processor is not in VMX mode, the TLBVMX word designating a field in a control register.

42. (Original) The method of claim 38 further comprising:

asserting the VMX tag word for a new TLB entry when the processor is not in VMX mode, the VMX tag word being a single bit;

negating the VMX tag word for a new TLB entry when the processor is in VMX mode; and

invalidating the TLB entry when an invalidation operation is performed and the VMX tag word is negated and the processor is in VMX mode.

43. (Original) The method of claim 42 wherein invalidating comprises invalidating the TLB entry irrespective of the value of the VMX tag word when an invalidation operation is performed and the processor is not in VMX mode.

44. (Original) The method of claim 42 wherein invalidating comprises invalidating the TLB entry when an invalidation operation is performed and the VMX tag word matches the TLBVMX word and the processor is not in VMX mode, the TLBVMX word designating a field in a control register.

45. (Original) The method of claim 42 wherein invalidating comprises invalidating the TLB entry by the invalidation operation conditioned upon value of one or more control words associated with the TLB.

46. (Original) The method of claim 45 wherein the control word or words associated with the TLB are located in one or more of the control registers of the processor or in a Virtual Machine Control Structure (VMCS) in memory.

47. (Original) The method of claim 46 further comprising distinguishing one of the control words associated with the TLB such that the VMX tag word is set to match a distinguished control word for a new TLB entry,

the distinguished control word associated with the TLB being designated the TLBVMX word.

48. (Original) The method of claim 47 further comprising setting the TLBVMX word to one of a plurality of values that constitute a proper subset of a set of all possible values for a VMX tag word when the processor mode corresponds to execution in a virtual machine (VM).

49. (Original) The method of claim 48 wherein invalidating comprises invalidating the TLB entry when an invalidation operation is performed and the value of the associated VMX tag word matches the value of the TLBVMX word and the processor mode corresponds to execution in a virtual machine (VM).

50. (Original) The method of claim 49 further comprising configuring the one or more control words associated with the TLB, including the TLBVMX word, when the processor mode corresponds to execution not in a virtual machine.

51. (Original) The method of claim 48 further comprising setting the TLBVMX word to one of a plurality of values that constitute a proper subset of a set of all possible values for a VMX tag word when the processor mode corresponds to execution not in a virtual machine.

52. (Original) The method of claim 51 wherein a set theoretic intersection of the plurality of values allowable for the TLBVMX word when the processor mode corresponds to execution not in a virtual machine and the plurality of values allowable for the TLBVMX word when the processor mode corresponds to execution in a virtual machine (VM) is an empty set.

53. (Original) The method of claim 51 wherein invalidation of a TLB entry by an invalidation operation is further conditioned upon value of a second control word associated with the TLB when the processor mode corresponds to execution not in a virtual machine.

54. (Original) The method of claim 53 wherein invalidating comprises invalidating the TLB entry when an invalidation operation is performed and the associated VMX tag word matches the second control word associated with the TLB and the processor mode corresponds to execution not in a virtual machine.

55. (Original) The method of claim 53 wherein invalidating comprises invalidating the TLB entry when an invalidation operation is performed and the logical AND of the associated VMX tag word and the second control word associated with the TLB matches the TLBVMX word and the processor mode corresponds to execution in a virtual machine, the second control word associated with the TLB being designated the TLBVMX mask word.

56. (Original) The method of claim 55 further comprising accessing by software to configure a portion of the TLBVMX word when the processor mode corresponds to execution in a virtual machine (VM) conditioned upon value of the TLBVMX mask word such that software executing when the processor mode corresponds to execution in a virtual machine (VM) is able to set the TLBVMX word to a value such that the logical AND of a new value of the TLBVMX word and a value of the TLBVMX mask word matches a logical AND of a previous value of the TLBVMX word and the value of TLBVMX mask word.

57. Cancel.

58. Cancel.

59. Cancel.

60. Cancel.

61. Cancel.

62. (Original) The method of claim 53 wherein invalidating comprises invalidating the TLB entry when an invalidation operation is performed and the logical AND of the associated VMX tag word with a logical NOT of the second control word associated with the TLB matches the TLBVMX word and the processor mode corresponds to execution in a virtual machine (VM), the second control word associated with the TLB being designated the TLBVMX inverted mask word.

63. (Original) The method of claim 62 further comprising accessing by software to configure a portion of the TLBVMX word when the processor mode corresponds to execution in a virtual machine (VM) conditioned upon the value of the TLBVMX inverted mask word such that software executing when the processor mode corresponds to execution in a virtual machine (VM) is able to set the TLBVMX word to a value such that a logical AND of a new value of the TLBVMX word and the logical NOT of the TLBVMX inverted mask word matches the logical AND of a previous value of the TLBVMX word and the logical NOT of TLBVMX inverted mask word.

64. Cancel.

65. Cancel.

66. Cancel.

67. Cancel.

68. Cancel.

69. (Original) The method of claim 51 wherein determining comprises determining size of a VMX tag word and the TLBVMX word by executing a specified processor instruction in a specified manner.

70. (Original) The method of claim 69 wherein the processor is compatible with the Intel Architecture and the specified instruction is a CPUID instruction and the specified manner is to have a specified value in an EAX register when the CPUID instruction is executed.

71. (Currently Amended) A processor comprising:

a translation lookaside buffer (TLB) having a plurality of TLB entries, each TLB entry being associated with a virtual machine extension (VMX) tag word indicating if the associated TLB entry is invalidated according to the processor mode when an invalidation operation is performed, the processor mode being one of execution in a virtual machine (VM) and execution not in a virtual machine, the invalidation operation belonging to a non-empty set of invalidation operations composed of a union of (1) a possibly empty set of operations that invalidate a variable number of TLB entries, (2) a possibly empty set of operations that invalidate exactly one TLB entry, (3) a possibly empty set of operations that invalidate the plurality of TLB entries, (4) a possibly empty set of operations that enable and disable use of virtual memory, and (5) a possibly empty set of operations that configure physical address size, page size or other

virtual memory system behavior in a manner that changes the manner in which a physical machine interprets the TLB entries; and

first, second, and third registers coupled to the TLB to store information related to the invalidation operation;

wherein the invalidation operations include IA-32-specific operations (a) IA-32 task switches involving changes of virtual memory related control registers, or (b) loading of control registers that modify IA-32-specific page size extension (PSE) and physical address extension (PAE).

72. (Currently Amended) The processor of claim 71 wherein the invalidation operation is one of (1) a loading of the first control register conditioned on a global bit, (2) an execution of a page invalidate instruction, (3) an IA-32 task switch involving change of at least one virtual memory related control register ~~task switch that modifies the first control register, (4) a loading of the second control register that modifies one of a protected mode indicator and a page mode indicator~~, and (5) a loading of the third control register that modifies one of an IA-32-specific page size extension (PSE), a page global enable (PGE), and a physical address extension (PAE).

73. (Original) The processor of claim 72 wherein the processor is in or not in VMX mode and the TLB entry is not invalidated at loading of the first control register when one of a transition into VMX mode (a VM entrance) and a transition out of VMX mode (a VM exit) occurs.

74. (Original) The processor of claim 73 wherein the VMX tag word is a single bit and

the VMX tag word is negated for a new TLB entry when the processor is not in VMX mode and the VMX tag word is asserted for a new TLB entry when the processor is in VMX mode; and

the TLB entry is invalidated when an invalidation operation is performed and the VMX tag word is asserted and the processor is in VMX mode.

75. (Original) The processor of claim 74 wherein the TLB entry is invalidated irrespective of value of the VMX tag word when an invalidation operation is performed and the processor is not in VMX mode.

76. (Original) The processor of claim 74 wherein a field in a control register is designated the translation lookaside buffer virtual machine extension (TLBVMX) word and the TLB entry is invalidated when an invalidation operation is performed and the VMX tag word matches the TLBVMX word and the processor is not in VMX mode.

77. (Original) The processor of claim 73 wherein the VMX tag word is a single bit and

the VMX tag word is asserted for a new TLB entry when the processor is not in VMX mode and the VMX tag word is negated for a new TLB entry when the processor is in VMX mode; and

the TLB entry is invalidated when an invalidation operation is performed and the VMX tag word is negated and the processor is in VMX mode.

78. (Original) The processor of claim 77 wherein the TLB entry is invalidated irrespective of the value of the VMX tag word when an invalidation operation is performed and the processor is not in VMX mode.

79. (Original) The processor of claim 77 wherein a field in a control register is designated the TLBVMX word and the TLB entry is invalidated when an invalidation operation is performed and the VMX tag word matches the TLBVMX word and the processor is not in VMX mode.

80. (Canceled) ~~The processor of claim 71 wherein invalidation of a TLB entry by an invalidation operation is further conditioned upon value of one or more control words associated with the TLB.~~